June 21-22, 2006

PDS Validation Tool Code Walk-through – Review of the architecture, design and code

Attendees: Atmospheres (Nancy Chanover, Lyle Huber, Joni Johnson), Engineering (Sean Hardman, Paul Ramirez, Mike Cayanan, Emily Law, Elizabeth Rye, Dan Crichton, Sean Kelly), Imaging (Myche McAuley), Program Management (Al Schultz), NASA Ames (Mark Rose), NASA HQ (Bill Knopf, Patrick Clark, Greg Gallant)

The following general comments were captured during the two days of meetings.

- Extension of standards to support international characters

  The issue is whether we will need to extend the PDS standards to support an extended character set for capturing metadata in other languages.

- Why not XML?

  The reviewers stressed the capabilities of XML and questioned why PDS was not using XML. The response was that that ODL predates XML and is used by the community, but that from an engineering perspective XML can be moved under the hood. In addition, it was discussed that PDS 4 might be the appropriate place for broader discussions on XML.

- Label Design

  A discussion ensued on label design and the need for tools. Bill recommended getting the process for label design moving along right behind validation tools so there is no loss of momentum.

- Data Dictionary Updates

  Elizabeth is going to convene a series of meetings so that the tool developers can provide feedback on standards issues.

- Code Coverage Testing

  A question was raised as to whether or not "code coverage" testing would be completed. Mark Rose suggested several open source tools that could be used for this purpose.

- Iterative Regression Approach

The reviewers suggested that the testing include a feedback mechanism to update the regression tests as new tests are identified. This is consistent with the EN plan.

- Capture test for each bug

  It was recommended that unit tests be updated as bugs are found to include tests for those bugs. This is consistent with the EN unit test plan.

- Requirements/Test Matrix

  It was recommended that a matrix be constructed that includes a mapping from the requirements to the tests that are being performed to check those requirements. In addition, it was recommended that this matrix be posted at EN and that the nodes be sent a link to it.

- Benchmarking between LVTOOL and VTOOL

  It was recommended that some benchmarking be done to compare VTOOL to LVTOOL. This includes performance tests, validation tests, reporting tests, etc. This will help demonstrate improvements.

- Strategy of Testing

  A question was raised as to the overall strategy. It was reported that the nodes will be involved in helping to define and execute the test cases. Bill agreed with this strategy.

The following code related comments were captured during the code-walkthrough on the first day:

- A suggestion from Mark to change the package name to gov.nasa.pds.tools… overall.
- Mark pointed out an inconsistency with package naming (label.validate and object.validator).
- Mark suggested that the code can be attached to the ANTLR parse tree instead of in separate classes. Mark and Paul will discuss this separately.

- pds-tools-0.1.0/src/resources/grammar/**odl.g**
  - Mark suggested that we make our grammar available for documentation purposes. Sean K. suggested the BNF format.
  - The current grammar requires the end_object identifier. Elizabeth says that this optional. Need to reconcile.
  - We should include line boundaries in the grammar.
  - Can the comment start on the same line as a statement? Yes, if it is after the statement.

- Mark wanted to know whether we are opening up streams as we evaluate pointers or are we doing it after the fact. Yet to be determined.
- Lyle pointed out that Groups currently allow all statements when they should be limited to attribute statements. Reconcile with standards.
- Units should be defined with a question mark, assume that means optional.
- Typo “seperator” should be “separator”.
- The grammar does not catch the fact that sets cannot contain sequences. This will be incorporated later.
- Line 150 and 196 the EOL is commented out. We need to resolve this item.
- Line 172 it appears that the code allows an identifier to start with a slash.
- Line 176, is white space allowed inside the angle brackets for units?
- Line 193, the code allows a double quote inside single quotes, inside a double quoted string. This should not be allowed according to the standards.
- Need to make sure the grammar does not allow statements after a comment and multiple statements per line.
- Paul would like to sit down with Elizabeth et al, to nail down the grammar with regard to the standards.

- pds-tools-0.1.0/src/java/gov/nasa/jpl/pds/tools/dict/**Dictionary.java**
  - Sean K. would like to see voluminous javadocs. Each class and method description should start with a summary of functionality.
  - Line 39, include the direction of the merge for the dictionaries.
  - Line 54, Mark pointed out a typo in the comment.
  - Line 59, Sean K. suggested using the contains() method.
  - Need to talk to Sean K. regarding refactoring related to Java 1.5.
  - Line 173, guard against null string.
  - Line 190 and 202, Sean K. suggested using Collections instead of Lists for reuse purposes.
  - Line 214, Mark suggested a reworking of the algorithm for the findObjectClassDefinition() method.

- pds-tools-0.1.0/src/java/gov/nasa/jpl/pds/tools/dict/**Definition.java**
  - Sean K. would like to see equals() and hashcode() methods for any class that is going to go into a container.
  - Sean K. suggested that we guard against null and that we use a consistent approach for dealing with nulls, either not pass them in or always check for them.

- pds-tools-0.1.0/src/java/gov/nasa/jpl/pds/tools/dict/**ElementDefinition.java**
  - Mark suggested storing the minimum and maximum as a numeric after it is parsed as a string.
  - Sean K. suggested that the code guard against negative length.

- pds-tools-0.1.0/src/java/gov/nasa/jpl/pds/tools/dict/**GroupDefinition.java**

    - Line 74 and 83, Mark noted that methods don't match bean patterns, meaning they don't start with is*.
    - Sean K. feels that canHaveElement() and hasElement() are the same functionality. Remove the long one.

- pds-tools-0.1.0/src/java/gov/nasa/jpl/pds/tools/dict/**ObjectDefinition.java**
    - Line 23, typo in comment.
    - Same comment regarding same functionality in GroupDefinition.java.

- pds-tools-0.1.0/src/java/gov/nasa/jpl/pds/tools/dict/parser/**DictionaryParser.java**
    - Mark suggested to instantiate the logger one way, the common way is as a static. Rely on the properties file specified by the client. Also remove the setLogger() method. Resolved with the configuration file.
    - Line 59, looks like an infinite loop.
    - Line 72, 76, 79 and 82, strings should be converted to tokens.
    - The code should not send stuff to stdout.
    - Line 92, uncomment.
    - Line 107, should break into two exceptions, or otherwise fix.

- pds-tools-0.1.0/src/java/gov/nasa/jpl/pds/tools/dict/parser/**DefinitionFactory.java**
    - Sean K. would prefer a map and factory method instead of the if/else statements.
    - Line 57, 125 and 169, fix the null exception case if NAME is not available.
    - Line 55, Sean K. suggested switching the if statement, which will alleviate the need for null check.

- pds-tools-0.1.0/src/java/gov/nasa/jpl/pds/tools/dict/type/**TypeCheckerFactory.java**
    - Line 25 and 32, should have static in the line.
    - Singletons use getInstance() where non-singletons would use newInstance().
    - Replace the big if/else block with a map.

- pds-tools-0.1.0/src/java/gov/nasa/jpl/pds/tools/label/**Label.java**
    - Line 169, Mark suggested synchronizing the get and set method names for label type.
    - Why does the code use separate maps instead of a single map? Need to resolve.
    - Line 146, no check for uniqueness of a statement.
    - Line 155, needs one more else with an exception throw.

- pds-tools-0.1.0/src/java/gov/nasa/jpl/pds/tools/label/**Statement.java**
    - Add equals() and hashcode() methods.
    - Line 66, possible null issue. Add a unit test.
    - Mark suggested possibly capturing column numbers.
    - Mark suggested having another class of value, which includes units.

    - Line 235, Mark suggested considering not using the Number format.
    - Line 241, remove printStackTrace().

- pds-tools-0.1.0/src/java/gov/nasa/jpl/pds/tools/label/**ObjectStatement.java**
    - Line 60, why are the parameters, List objects instead of a Map? Just look at it, but Sean K. thinks it is okay.
    - Line 130, add an else statement to throw illegal statement exception.

- pds-tools-0.1.0/src/java/gov/nasa/jpl/pds/tools/label/**GroupStatement.java**
    - Be consistent with convenience methods.

- pds-tools-0.1.0/src/java/gov/nasa/jpl/pds/tools/label/**AttributeStatement.java**
    - Line 51, add comment about using -1 as a default.
    - Mark suggested dealing with namespace in the constructor.

- pds-tools-0.1.0/src/java/gov/nasa/jpl/pds/tools/label/**Value.java**
    - Change wrapper to marker in the comment.
    - Sean K. suggested that this be a base class instead of an interface.

- pds-tools-0.1.0/src/java/gov/nasa/jpl/pds/tools/label/**Scalar.java**
    - Line 31, Sean K. suggested renaming value() to getValue().

- pds-tools-0.1.0/src/java/gov/nasa/jpl/pds/tools/label/**Numeric.java**
    - Line 80 and 96, rename methods with get*.

- pds-tools-0.1.0/src/java/gov/nasa/jpl/pds/tools/label/parser/**DefaultMessageListener.java**
    - Remove.

- pds-tools-0.1.0/src/java/gov/nasa/jpl/pds/tools/label/parser/**DefaultParserListener.java**
    - Remove.

- pds-tools-0.1.0/src/java/gov/nasa/jpl/pds/tools/label/parser/**LabelParserFactory.java**
    - Line 28, replace the call to newInstance() with getInstance().

- pds-tools-0.1.0/src/java/gov/nasa/jpl/pds/tools/label/validate/**DataObjectValidator.java**
    - Remove get and setLogger() methods.

- pds-tools-0.1.0/src/java/gov/nasa/jpl/pds/tools/label/validate/**DataObjectValidatorFactory.java**
    - Mark suggested not to use System.getProperties(), System.out() and System.err() for classes intended for web applications.

    - Line 30, fix the newInstance() call and synchronize.

- pds-tools-0.1.0/src/test/gov/nasa/jpl/pds/tools/ddict/**GroupDefinitionTest.java**
    - Mark suggested the need to test for invalid or optional elements.

- pds-tools-0.1.0/**build.xml**
    - Check out the bullet symbols in this auto-generated file.